7u

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/736,436	12/15/2003	Dan Li	CIS03-50(7980)	3432

22468   7590   06/19/2007
CHAPIN & HUANG L.L.C.
WESTBOROUGH OFFICE PARK
1700 WEST PARK DRIVE
WESTBOROUGH, MA 01581

EXAMINER
GAFFIN, JEFFREY A

ART UNIT	PAPER NUMBER
2165	

MAIL DATE	DELIVERY MODE
06/19/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

	Application No.	Applicant(s)
	10/736,436	LI ET AL.
Office Action Summary	Examiner	Art Unit	
	Fariborz Khoshnoodi	2168	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE <u>3</u> MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.
- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). .
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) ☒ Responsive to communication(s) filed on <u>15 December 2003</u>.

2a) ☐ This action is **FINAL**.   2b) ☒ This action is non-final.

3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4) ☒ Claim(s) <u>1-22</u> is/are pending in the application.

    4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5) ☐ Claim(s) _____ is/are allowed.

6) ☒ Claim(s) <u>1-22</u> is/are rejected.

7) ☐ Claim(s) _____ is/are objected to.

8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9) ☐ The specification is objected to by the Examiner.

10) ☒ The drawing(s) filed on <u>12/15/2003</u> is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.

    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

    a) ☐ All b) ☐ Some * c) ☐ None of:

      1. ☐ Certified copies of the priority documents have been received.

      2. ☐ Certified copies of the priority documents have been received in Application No. _____.

      3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

    * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO/SB/08)
    Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413)
    Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

Detailed Action

Claim Rejections - 35 USC § 102

1.   The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the

basis for the rejections under this section made in this Office Action:

> A person shall be entitled to a patent unless -
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed
> in the United States before the invention by the applicant for patent or (2) a patent granted on an application for
> patent by another filed in the United States before the invention by the applicant for patent, except that an
> international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this
> subsection of an application filed in the United States only if the international application designated the United
> States and was published under Article 21(2) of such treaty in the English language.

2.   Claims 1-22 are rejected under 35 U.S.C. § 102(e) as being anticipated by Hefetz et al.

United States Patent Publication No. 20040123238 A1.

As per claim 1:

        Hefetz et al. teach a method comprising: **receiving a request for the portal from a**

client system *(i.e., "The portal 220 receives requests from the*

clients 200 and uses portal templates to generate information

views 225 (e.g., web pages) in response." (Par. 33 lines 2-4));

accessing a portal template in response to the request, the portal template having at least

one dynamic portion *(i.e., "First, the scripting variable can enable*

differentiating code segments according to the requesting user

agent (e.g., browser type, version, etc.). A portal UserAgent

service can be accessed to detect the user agent in use.

Second, the scripting variable can be used to resolve whether

the JSP layout template is currently used for run-time or

design-time purposes, and can condition code accordingly."(Par.

55)); **including into the at least one dynamic portion of the portal template links to**

content cached in the content engine and information about content availability to generate

a portal page *(i.e., "A portal brings together various applications*

from an intranet and an extranet that may or may not be·related

to one another. Traditional portal software products have

included portal development tools that allow creation of portal

templates to be used at portal run-time to generate portal pages

for display. Run-time portal templates, such as may be

implemented using Java Server Pages (JSP) and custom tag

libraries, provide an efficient way to combine static data with

dynamic run-time data for presentation to a user in a portal

environment."(Par. 3)); **and providing the portal page to the client system** *(i.e.,*

"The portal 220 can implement a user-roles based system to

personalize the common interface and the information views 225

for a user of a client 200."(Par. 33 lines 4-7)).

As per claim 2:

Hefetz et al. teach a method, **wherein including information about content availability**

further comprises the steps of comparing a replication status to a catalog of files carried in

the content engine to determine what files are locally cached and what files remain to be

downloaded *(i.e., "The JSP template 660 defines the general*

structure/design of the page layout, and page layout definitions

for the JSP template 660 can be stored in a PCD (Portal Content

Directory--metadata repository implementation of the portal)

630. A portal component profile for the layout component 670

can represent the page layout definitions as stored in the PCD

630, and/or the JSP template 660 can be stored in the PCD

630." (Par. 56 lines 7-14)); **and writing a list of files that remain to be**

downloaded to the portal page with an indicator of unavailability *(i.e., "For*

example, a portal developer may create a template with two

iViews: a first iView on the left in a narrow column with a list

of items to select, and a second iView on the right in a wide

column with details of a current item selected in the list. The

portal developer can readily select which components to place on

a page, set permissions and/or attributes for user-specific

personalization, specify the layouts of multiple portal pages by

defining the portal templates in the GUI that presents visual

representations of the portal pages to be generated at run-time

using the templates, and set the structure of the content

components in the templates." (Par. 45 lines 4-15)).

As per claim 3:

Hefetz et al. teach a method, **wherein the step of including links to content cached in the content engine further comprises the steps of: checking a replication status of the content engine to determine available cached content** *(i.e., "The WYSIWYG page editor can also include a list of all run-time content-presentation components currently available in the portal system, which can be dragged into and dragged out of content containers as desired in the GUI of the WYSIWYG page editor." (Par. 28 lines 14-18));* **and including into the at least one dynamic portion of the portal template links to content found in the replication status** *(i.e., "The portal 220 can be a portal software product that includes out-of-the-box portal templates and portal development tools that can be used to create portal templates. These tools can be used by a portal developer and/or administrator to design and deploy a portal in a particular IT environment, and these tools include a WYSIWYG portal page editor. Dynamic content components of a portal page can be specified in a portal template using selectively interpreted content placeholders." (Par. 38 lines 1-9)).*

As per claim 4:

Hefetz et al. teach a method further comprising **the step of hiding at least one link to**

content not found in the replication status in the at least one dynamic portion of the portal template *(i.e., "The layout component 670 ·calls a content container tag handler 650 to pull the dynamic content from the main content storage 640 and place the dynamic content in specified position(s) in the page. For example, the tag handler 650 can call the ILayoutStructure 620 to get a list of the iViews it should include, and then turn to the main content storage 640 to get the iViews content to include in the page. With the dynamic content added, a final portal page 670 can then be sent to the page requester."(Par.58))*.

As per claim 5:

Hefetz et al. teach a method, **wherein the portal template includes at least one applet and the step of including links into the portal template comprises running the at least one applet to acquire at least one pointer to content cached in the content engine** *(i.e., "The defined page element can be a custom Java Server Page tag, the design-time translator and the run-time translator can be Java Server Page tag handlers for the custom Java Server Page tag, and the run-time translator can obtain portal dynamic content according to the portal page template whereas the design-time translator need not do so."(Par. 9 lines 9-14))*.

As per claim 6:

Hefetz et al. teach a method further comprising **the step of providing the portal template having at least one applet to the client system and wherein the client system instantiates the portal template including at least one applet and executes the at least one applet to acquire content cached in the content engine** *(i.e., "Invoking the design-time translator can also result in client-side scripting components being included in the representation to form at least part of the design-time application and enable adding a content component to a content container in the portal page template using a drag-and-drop action. As before, the defined page element can be a custom Java Server Page tag, the design-time translator and the run-time translator can be Java Server Page tag handlers for the custom Java Server Page tag, and the run-time translator can obtain portal dynamic content according to the portal page template, whereas the design-time translator need not do so." (Par. 12))*.

As per claim 7:

Hefetz et al. teach a method, **wherein the step of accessing the portal template further comprises reading a template stored in the content engine** *(i.e., "The enterprise portal can include a portal platform in communication with a navigation platform accessed by the user. The portal platform*

can include a web server, a page builder, and an information

view (AKA, integration view or iView) server. The portal

platform can also include a unification server, user management

components (e.g., a Corporate LDAP (Lightweight Directory Access

Protocol) Directory and a Portal LDAP Directory), and a database

repository. The database repository can include an SQL

(Structured Query Language) Database and a Portal Content

Directory (PCD)."(Par. 36 lines 7-17)).

As per claim 8:

Hefetz et al. teach a method, **wherein the step of accessing the portal template further**

comprises accessing a template stored at a portal page server in the content distributed

network (i.e., "FIG. 2 illustrates a portal-based networked

environment. Multiple clients 200 can access data over a

network 210 through a portal 220. The network 210 can be any

communication network linking machines capable of communicating

using one or more networking protocols, e.g., a local area

network (LAN), a wide area network (WAN), an enterprise network,

a virtual private network (VPN), a mobile device network and/or

the Internet. The clients 200 can be any machines or processes

capable of ommunicating over the network 210. The clients 200

can be web browsers and can be communicatively coupled with the

network 210 through a proxy server."(Par. 32)).

As per claim 9:

Hefetz et al. teach a method, **wherein the request is a redirected request from the**

client system, redirected away from a central site and to the content engine by a content

router in the content distributed network *(i.e., "An enterprise portal and*

associated enterprise base systems can reside in one or more

programmable machines, which can communicate over a network or

one or more communication busses. For example, the enterprise

portal can reside in one or more servers connected to a public

network and can run one or more server software programs. The

enterprise portal can include a portal platform in communication

with a navigation platform accessed by the user. The portal

platform can include a web server, a page builder, and an

information view (AKA, integration view or iView) server. The

portal latform can also include a unification server, user

management components (e.g., a Corporate LDAP (Lightweight

Directory Access Protocol) Directory and a Portal LDAP

Directory), and a database repository. The database repository

can include an SQL (Structured Query Language) Database and a

Portal Content Directory (PCD)." (Par. 36)).

As per claim 10:

Hefetz et al. teach a method, **wherein the request is a search request and the method**

further comprises the steps of: querying a central server in response to the search request

(i.e., "Moreover, the enterprise portal can include, or be in

communication with, various other enterprise systems, such as a

knowledge management platform, a text retrieval and extraction

server, and a business warehouse platform. The knowledge

management platform can be software for data preparation,

including workflow, publishing, feedback, analysis, discussion,

querying, indexing, profiling, concurrency control, and

classification functionality."(Par. 37 lines 3-11)); **and receiving a**

list of files in response to querying the central server *(i.e., "Portal development*

tools can be used to edit portal pages and tailor a portal to an

organization and its information technology (IT) environment.

Editing portal pages generally involves placing content

omponents on the pages.. In conventional portal development

tools, this editing occurs in a user interface (UI) that

provides a schematic representation of a portal page, including

a schematic list of containers that hold run-time content."(Par.

4 lines 1-8)); **and wherein the step of including links to content further comprises**

including links to files from the list cached in the content engine *(i.e., "A portal*

development tool can provide a GUI (graphic user interface)

WYSIWYG portal template editor. The same portal template can be

used by the template editor at design-time, while the template

is being created, and by a server at run-time, when the template

is deployed to portal users. Components of the portal template

can be rendered the same at design-time as they are at run-time,

with the exception that, at design-time, portal dynamic content

in content containers can be replaced by a representation of the

dynamic content." (Par. 6)).

As per claim 11:

Hefetz et al. teach a method, **wherein the request is received at the content engine**

based on a network location of the content engine with respect to the client system

(i.e., "FIG. 2 illustrates a portal-based networked environment.

Multiple clients 200 can access data over a network 210 through

a portal 220. The network 210 can be any communication network

linking machines capable of communicating using one or more

networking protocols, e.g., a local area network (LAN), a wide

area network (WAN), an enterprise network, a virtual private

network (VPN), a mobile device network and/or the

Internet." (Par. 32 lines 1-8)).

As per claim 12:

Hefetz et al. teach a method comprising: **receiving a request for the channel portal**

from a client system *(i.e., "The portal 220 receives requests from the*

clients 200 and uses portal templates to generate information

views 225 (e.g., web pages) in response." (Par. 33 lines 2-4)); **accessing a channel portal template in response to the request, the channel portal template having at least one dynamic portion** *(i.e., "First, the scripting variable can enable differentiating code segments according to the requesting user agent (e.g., browser type, version, etc.). A portal UserAgent service can be accessed to detect the user agent in use. Second, the scripting variable can be used to resolve whether the JSP layout template is currently used for run-time or design-time purposes, and can condition code accordingly." (Par. 55));* **including into the at least one dynamic portion of the channel portal template links to content cached in the content engine and information about content availability to generate a channel portal page** *(i.e., "A portal brings together various applications from an intranet and an extranet that may or may not be related to one another. Traditional portal software products have included portal development tools that allow creation of portal templates to be used at portal run-time to generate portal pages for display. Run-time portal templates, such as may be implemented using Java Server Pages (JSP) and custom tag libraries, provide an efficient way to combine static data with dynamic run-time data for presentation to a user in a portal environment." (Par. 3));* **and providing the channel portal page to the client system** *(i.e., "The portal 220*

can implement a user-roles based system to personalize the

common interface and the information views 225 for a user of a

client 200."(Par. 33 lines 4-7)).

As per claim 13:

Hefetz et al. teach a method comprising: **checking a replication status of the content**

engine to determine channel content available at the content engine *(i.e., "The*

WYSIWYG page editor can also include a list of all run-time

content-presentation components currently available in the

portal system, which can be dragged into and dragged out of

content containers as desired in the GUI of the WYSIWYG page

editor."(Par. 28 lines 14-18)); **and including into the at least one dynamic**

portion of the channel portal template links to channel content found in the replication

status to generate the channel portal page *(i.e., "The portal 220 can be a*

portal software product that includes out-of-the-box portal

templates and portal development tools that can be used to

create portal templates. These tools can be used by a portal

developer and/or administrator to design and deploy a portal in

a particular IT environment, and these tools include a WYSIWYG

portal page editor. Dynamic content components of a portal page

can be specified in a portal template using selectively

interpreted content placeholders."(Par. 38 lines 1-9)).

As per claim 14:

Hefetz et al. teach a method, **wherein the request includes a search query for content in the channel, wherein the channel portal template includes an applet accepting a first input of the search query and a second input of a list of content in the channel and wherein the step of including links to content further includes the steps of: executing the applet to find content matching the search query** *(i.e., "The portal 220 can be a portal software product that includes out-of-the-box portal templates and portal development tools that can be used to create portal templates. These tools can be used by a portal developer and/or administrator to design and deploy a portal in a particular IT environment, and these tools include a WYSIWYG portal page editor. Dynamic content components of a portal page can be specified in a portal template using selectively interpreted content placeholders." (Par. 38 lines 1-9))*; **determining whether the content matching the search query is cached at the content engine** *(i.e., "A portal development tool can provide a GUI (graphic user interface) WYSIWYG portal template editor. The same portal template can be used by the template editor at design-time, while the template is being created, and by a server at run-time, when the template is deployed to portal users. Components of the portal template can be rendered the same at design-time as they are at run-time, with the exception that, at design-*

time, portal dynamic content in content containers can be replaced by a representation of the dynamic content."(Par. 6)); **and including into the at least one dynamic portion of the channel portal template links to channel content cached at the content engine** (i.e., "A portal brings together various applications from an intranet ·and an extranet that may or may not be related to one another. Traditional portal software products have included portal development tools that allow creation of portal templates to be used at portal run-time to generate portal pages for display. Run-time portal templates, such as may be implemented using Java Server Pages (JSP) and custom tag libraries, provide an efficient way to combine static data with dynamic run-time data for presentation to a user in a portal environment."(Par. 3)).

As per claim 15:

Hefetz et al. teach a system comprising: **a network interface to receive a request for a portal from a client system** *(i.e., "The portal 220 provides a common interface to applications 240. The portal 220 receives requests from the clients 200 and uses portal templates to generate information views 225 (e.g., web pages) in response."(Par. 33 lines 1-4))*; **a storage device to store content from the content distributed network and a portal template having at least one dynamic portion** *(i.e., "The dynamic run-*

time content can be gathered in parallel (e.g., by using java multi-threading) to improve performance. For example, the iViews content for the page can be fetched from an iViews content gathering resource 610, which can be one or more program components that gather iViews content for the page in parallel (e.g., an iView server). The fetched content blocks can be stored in a main content storage 640 by the page builder 600 as they are obtained. The fetched dynamic content can be stored in blocks with no ordering, such as in an array of iViews' content."(Par. 57)); **and a controller coupled to the interface and the storage device, the controller configured to access the portal template in response to the request, to include in the at least one dynamic portion of the portal template links to content cached in the content engine and information about content availability to generate a portal page, and to provide the portal page to the client system** *(i.e., "The portal developer can readily select which components to place on a page, set permissions and/or attributes for user-specific personalization, specify the layouts of multiple portal pages by defining the portal templates in the GUI that presents visual representations of the portal pages to be generated at run-time using the templates, and set the structure of the content components in the templates."(Par. 45 lines 8-15)).*

As per claim 16:

Hefetz et al. teach a method, **wherein the storage device further includes a replication status of the content engine and the controller is further configured to check the replication status to determine available cached content, the controller further to include into the at least one dynamic portion of the portal template links to content found in the replication status** *(i.e., "The portal 220 can be a portal software product that includes out-of-the-box portal templates and portal development tools that can be used to create portal templates. These tools can be used by a portal developer and/or administrator to design and deploy a portal in a particular IT environment, and these tools include a WYSIWYG portal page editor. Dynamic content components of a portal page can be specified in a portal template using selectively interpreted content placeholders." (Par. 38 lines 1-9)).*

As per claim 17:

Hefetz et al. teach a method, **wherein the portal template includes at least one applet and the controller is further configured to run the at least one applet to acquire at least one pointer to content cached in the content engine** *(i.e., "The defined page element can be a custom Java Server Page tag, the design-time translator and the run-time translator can be Java Server Page tag handlers for the custom Java Server Page tag, and the run-*

time translator can obtain portal dynamic content according to

the portal page template whereas the design-time translator need

not do so."(Par. 9 lines 9-14)).

As per claim 18:

Hefetz et al. teach a method, **wherein the portal is a channel portal and the portal**

template is a channel portal template and the controller is further configured to include

into the at least one dynamic portion of the channel portal template links to content cached

in the content engine to generate a channel portal page *(i.e., "A portal*

development tool can provide a GUI (graphic user interface)

WYSIWYG portal template editor. The same portal template can be

used by the template·editor at design-time, while the template

is being created, and by a server at run-time, when the template

is deployed to portal users. Components of the portal template

can be rendered the same at design-time as they are at run-time,

with the exception that, at design-time, portal dynamic content

in content containers can be replaced by a representation of the

dynamic content."(Par. 6)).

As per claim 19:

Hefetz et al. teach a method, **wherein the storage device further stores a replication**

status of the content engine and the controller is further configured to check the replication

status to determine channel content available at the content engine and to include into the

at least one dynamic portion of the channel portal template links to channel content found

in the replication status to generate a channel portal page *(i.e., "The portal 220*

can be a portal software product that includes out-of-the-box

portal templates and portal development tools that can be used

to create portal templates. These tools can be used by a portal

developer and/or administrator to design and deploy a portal in

a particular IT environment, and these tools include a WYSIWYG

portal page editor. Dynamic content components of a portal page

can be specified in a portal template using selectively

interpreted content placeholders."(Par. 38 lines 1-9)).

As per claim 20:

Hefetz et al. teach a method, **wherein the request includes a search query for content**

in the channel, wherein the channel portal template includes an applet that accepts a first

input of the search query and a second input of a list of content in the channel, and wherein

the controller is further configured to execute the applet to find content matching the

search query, to determine whether the content matching the search query is cached at the

content engine, and to include into the at least one dynamic portion of the channel portal

template links to channel content cached at the content engine *(i.e., "The portal*

220 can be a portal software product that includes out-of-the-

box portal templates and portal development tools that can be

used to create portal templates. These tools can be used by a

portal developer and/or administrator to design and deploy a

portal in a particular IT environment, and these tools include a

WYSIWYG portal page editor. Dynamic content components of a

portal page can be specified in a portal template using

selectively interpreted content placeholders."(Par. 38 lines 1-

9)… "A portal development tool can provide a GUI (graphic user

interface) WYSIWYG portal template editor. The same portal

template can be used by the template editor at design-time,

while the template is being created, and by a server at run-

time, when the template is deployed to portal users. Components

of the portal template can be rendered the same at design-time

as they are at run-time, with the exception that, at design-

time, portal dynamic content in content containers can be

replaced by a representation of the dynamic content."(Par. 6)…

"A portal brings together various applications from an intranet

and an extranet that may or may not be related to one another.

Traditional portal software products have included portal

development tools that allow creation of portal templates to be

used at portal run-time to generate portal pages for display.

Run-time portal templates, such as may be implemented using Java

Server Pages (JSP) and custom tag libraries, provide an

efficient. way to combine static data with dynamic run-time data for presentation to a user in a portal environment."(Par. 3)).

As per claim 21:

Hefetz et al. teach a method comprising: **providing a manifest file to establish a channel of content in the content distributed network, the manifest file describing channel content, the manifest file further including a portal template, the portal template including at least one dynamic portion** *(i.e., "In another aspect, a technique, which can be implemented in a software product, involves translating a placeholder in a portal template, during design-time of a portal page, into a representation of a container designed to present portal dynamic content associated with the placeholder, and presenting a WYSIWYG portal layout editor using the representation of the container designed to present the portal dynamic content."(Par. 10 lines 1-8))* ; **providing at least one content engine to cache a portion of channel content and to cache the portal template** *(i.e., "Then, during run-time of a portal page, obtaining the portal dynamic content from a dynamic content source, and translating the placeholder in the portal template into a presentation of the container and the obtained portal dynamic content."(Par. 10 lines 8-12))*; **receiving at the at least one content engine a request for the Web portal from a client system** *(i.e., "The portal 220*

provides a common interface to applications 240. The portal 220

receives requests from the clients 200 and uses portal templates

to generate information views 225 (e.g., web pages) in

response."(Par. 33 lines 1-4)); **accessing by the at least one content engine the**

portal template in response to the request *(i.e., "First, the scripting*

variable can enable differentiating code segments according to

the requesting user agent (e.g., browser type, version, etc.).

A portal UserAgent service can be accessed to detect the user

agent in use. Second, the scripting variable can be used to

resolve whether the JSP layout template is currently used for

run-time or design-time purposes, and can condition code

accordingly." (Par. 55)); **including by the at least one content engine into the**

least one dynamic portion of the portal template links to content cached in the content

engine to generate a Web portal page *(i.e., "First, the scripting variable*

can enable differentiating code segments according to the

requesting user agent (e.g., browser type, version, etc.). A

portal UserAgent service can be accessed to detect the user

agent in use. Second, the scripting variable can be used to

resolve whether the JSP layout template is currently used for

run-time or design-time purposes, and can condition code

accordingly."(Par. 55)… "The portal 220 provides a common

interface to applications 240. The portal 220 receives requests

from the clients 200 and uses portal templates to generate

information views 225 (e.g., web pages) in response."(Par. 33

lines 1-4)); **and providing by the content engine the Web portal page to the client**

system *(i.e., "The portal 220 provides a common interface to*

applications 240. The portal 220 receives requests from the

clients 200 and uses portal templates to generate information

views 225 (e.g., web pages) in response. The portal 220 can

implement a user-roles based system to personalize the common

interface and the information views 225 for a user of a client

200"(Par. 33 lines 1-7)).

As per claim 22:

Hefetz et al. teach a **computer program product having a computer-readable medium**

including computer program logic encoded thereon that, when performed on a computer

system having a coupling of a memory, a processor, and at least one communications

interface, provides a method for dynamically providing a Web portal in a content

distributed network by performing the operations of: receiving a request for the portal

from a client system *(i.e., "The portal 220 provides a common*

interface to applications 240. The portal 220 receives requests

from the clients 200 and uses portal templates to generate

information views 225 (e.g., web pages) in response." (Par. 33

lines 1-4)); **accessing a portal template in response to the request, the portal template**

having at least one dynamic portion *(i.e., "The portal 220 can receive information 245 from the applications 240 to fulfill requests from the clients 200; this information can be dynamic content and the applications 240 can be dynamic content sources." (Par. 33 lines 9-13))*; **including into the at least one dynamic portion of the portal template links to content cached in the content engine and information about content availability to generate a portal page** *(i.e., "The portal developer can readily select which components to place on a page, set permissions and/or attributes for user-specific personalization, specify the layouts of multiple portal pages by defining the portal templates in the GUI that presents visual representations of the portal pages to be generated at run-time using the templates, and set the structure of the content components in the templates." (Par. 45 lines 8-15))*; **and providing the portal page to the client system** *(i.e., "The portal 220 provides a common interface to applications 240. The portal 220 receives requests from the clients 200 and uses portal templates to generate information views 225 (e.g., web pages) in response. The portal 220 can implement a user-roles based system to personalize the common interface and the information views 225 for a user of a client 200." (Par. 33 lines 1-7))*.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Fariborz Khoshnoodi whose telephone number is 571-270-1005. The examiner can normally be reached on M-Th every other F 8:00-4:00..

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tim Vo can be reached on 571-272-3642. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Fariborz Khoshnoodi
Examiner
Art Unit 2168 ·

FK

TIM VO
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100